



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/900,123	07/05/2001	Scott Wiltamuth	MSFT-0573/160076.1	5765
41505	7590	05/24/2010	EXAMINER	
WOODCOCK WASHBURN LLP (MICROSOFT CORPORATION)			VO, TED T	
CIRA CENTRE, 12TH FLOOR				
2929 ARCH STREET			ART UNIT	PAPER NUMBER
PHILADELPHIA, PA 19104-2891			2191	
			MAIL DATE	DELIVERY MODE
			05/24/2010	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No. 09/900,123	Applicant(s) WILTAMUTH ET AL.	
	Examiner TED T. VO	Art Unit 2191	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 19 February 2010.
- 2a) ☒ This action is **FINAL**. 2b) ☐ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1-3,6-15,17-19,23-26,28-37,39-41,61-63,66-75 and 77-79 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☒ Claim(s) 1-3,6-15,17-19,23-26,28-37,39-41,61-63,66-75 and 77-79 is/are rejected.
- 7) ☐ Claim(s) _____ is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on _____ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
- ☐ Certified copies of the priority documents have been received.
 - ☐ Certified copies of the priority documents have been received in Application No. _____.
 - ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- | | |
|---|---|
| 1) <input type="checkbox"/> Notice of References Cited (PTO-892) | 4) <input type="checkbox"/> Interview Summary (PTO-413) |
| 2) <input type="checkbox"/> Notice of Draftperson's Patent Drawing Review (PTO-948) | Paper No(s)/Mail Date. _____ |
| 3) <input type="checkbox"/> Information Disclosure Statement(s) (PTO/SB/08) | 5) <input type="checkbox"/> Notice of Informal Patent Application |
| Paper No(s)/Mail Date _____ | 6) <input type="checkbox"/> Other: _____ |

DETAILED ACTION

1. This action is in response to the communication filed on 02/19/2010.

Claims 77-79 are new. Claims 1-3, 6-15, 17-19, 23-26, 28-37, 39-41, 61-63, 66-75, 77-79 are pending.

Response to Arguments

2. This is in response to the arguments filed in the remarks filed on 02/19/2010.

Regarding to the argument to the rejection of claims 1-3, 6-15, 17-19, 23-26, 28-37, 39-41, 61-63, 66-75 under 35 U.S.C 103, as being unpatentable over Kramer: Applicants submit the disagreements on the claimed term, “explicit interfaces member” and the using of an explicit interface as shown in the references’ Figures based on the terms. Especially, Applicants submit Kramer uses a mechanism, its interface is only the object interacts with the method invocation, and Kramer fails to show the claimed phrase: : “[a]n *implemented explicit interface member to be excluded from a public interface of said class*”.

Examiner respectfully disagrees: From the claimed recitation, “*an explicit interface member mechanism that enables a class to implement an explicit interface member*”, it is claiming a programming statement implementation. In the specification, it introduces programming language syntax/rule implementation for a specific programming language. The programming language syntax/rule implementation appears being represented in a form, for

Art Unit: 2191

example: Interface abc { ... }; class abc {}, wherein a method or an instruction within the {} will be in an association with claiming. As noted that a programming statement implementation cannot be accountable for an infringement, because it is only a programming instruction per se. It is improper if one claim an infringement on programming statement implementation such as “If abc then {..} else {..}”, do abc {}, function abc {}, document.write.abc(), etc. Therefore, claiming programming statement implementation should be improper.

Applicants have attempted adding, “[c]onfiguring a computer to generate computer executable instructions” and, “[s]toring said class in a form that includes said implemented explicit interface member in a computer readable storage medium”. Although the amendment has put the programming instruction of the claims to be tangible in a storage medium, it diverts the claimed functionality as for storing a class in memory. It should be noted that the act of storing a class provides no patentability but a common thing in the art.

Regarding the Applicant’s arguments’ statements:

i. *“First, while the Examiner equates interfaces with "explicit interface member mechanism" and "explicit interface member," the Examiner fails to point out where this is taught in the cited reference”.*

- Examiner’s response: The citation is pointed out in the action.

ii. *“Second, Kramer's definition of "interfaces" suggests that an interface is only a set of methods and not an interface member. Kramer states: "Objects interact by method invocation. An object interface is usually described by those methods that it offers”.*

- Examiner’s response: The claimed recitation is drawn from a syntax rule

“interface abc {}; class abc {}”.

For example, depicting evidence as seen from the C# specification submitted early:

Art Unit: 2191

13.4.1 Explicit interface member implementations

For purposes of implementing interfaces, a class or struct may declare *explicit interface member*

implementations. An explicit interface member implementation is a method, property, event, or indexer declaration that references a fully qualified interface member name. For example

```
interface ICloneable
{
    object Clone();
}
interface IComparable
{
    int CompareTo(object other);
}
class ListEntry: ICloneable, IComparable
{
    object ICloneable.Clone() {...}
    int IComparable.CompareTo(object other) {...}
}
```

Here, ICloneable.Clone and IComparable.CompareTo are explicit interface member implementations.

This shows that an explicit interface member implementation is only a mere instruction such as a method, where in Kramer, it is also a particular programming implementation like an explicit interface member implementation in the C# specification.

iii. “Finally, Kramer also fails to teach or suggest the phrase “an implemented explicit interface member to be excluded from a public interface of said class” as stated in claim 1”.

- Examiner’s response: There is no proof for an instruction, which is within Kramer’s explicit interface, requiring as being included with *a public interface of a class*. The fact is that ‘being included or excluded’ is only a rule set forth in a certain compiler.

In general, the claimed invention shows that it is a method for “configuring” and “storing” a class into a storage medium where how this can be patentable is not addressed in the arguments.

Claim Rejections - 35 USC § 103

3. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

A person shall be entitled to a patent unless –

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

4. Claims 1-3, 6-15, 17-19, 23-26, 28-37, 39-41, 61-63, 66-75, 77-79 are rejected under 35 U.S.C. 103(a) as being unpatentable over Kramer et al., “Configuring Object-based Distributed Programs in REX” , Software Engineering Journal, 3-1992, pages 139-149.

As per claim 1: Kramer teaches *A method comprising:*

configuring a computer to generate computer executable instructions, using object-based computer code of an object-oriented programming language utilizing an explicit interface member mechanism that enables a class to implement an explicit interface member by explicitly specifying the relationship between the class and the explicit interface member, wherein the explicit interface member mechanism enables an implemented explicit interface member to be excluded from a public interface of said class; and storing said class in a form that includes said implemented explicit interface member in a computer readable storage medium.

Art Unit: 2191

- The claim, as a whole, is for configuring, utilizing an explicit interface member mechanism that enables a class, to implement an explicit interface member, and storing the class into a computer readable storage medium. Thus, it is very common; particularly, in the area of the storage device, where a storage device is for storing data such as software.

The reference teaches configuring by using a mechanism (See p. 140, title) that is well defined with interfaces (*explicit interface member mechanism*) to generate programming class (e.g. Fig. 4, p. 142). With this configuration language, in Fig. 4, it enables an object with explicit interfaces (*an explicit interface member*). Using the configuration language, the object with explicit interfaces will not require declaring “public” to the interfaces.

The reference does not address storing the class or the object with explicit interfaces “in a form” as emphasized in the claim with the feature “***storing said class in a form that includes said implemented explicit interface member in a computer readable storage medium***”.

However, storing a class in a form is merely a text that represents an instruction construct. This is the work done in for computer and in computer.

Therefore, it is obviously to ordinary in the art at the time to implement the text of this class into a file for being portable, where according to MPEP, make portable cannot be patentable over a reference.

As per claim 2: Kramer further teaches

A method according to claim 1, wherein said specifying of the relationship includes specifying a qualified name of the class. See p. 139, parent child class, p. 140, object naming, etc.

Art Unit: 2191

As per claim 3: Kramer further teaches *A method according to claim 2, wherein said specifying of the qualified name includes specifying an interface name and said at least one interface member name.* See p. 139, parent child class, and further see Figure 1, or 4.

As per claim 6: Kramer further teaches *A method according to claim 1, wherein the explicit interface member mechanism **enables** the class to implement an internal interface not accessible to a consumer of said class.* The bold phase is relative term that has the claim remain reciting “explicit interface member mechanism”. See sec. 2.2, start at p. 141, and refer to the type of ODD specified in the reference title.

As per claim 7: Kramer further teaches *A method according to claim 1, wherein said explicit interface member mechanism **enables disambiguation** of a plurality of interface members having the same signature.* The bold phase is relative term that has the claim remain reciting “explicit interface member mechanism”. Inherent in the term “explicit interface” as shown in Fig. 4, and see sec. 2.2, 2.3.

As per claim 8: Kramer further teaches *A method according to claim 1, wherein said explicit member mechanism **enables disambiguation** of a plurality of interface members having the same signature and return type.* The bold phase is relative term that has the claim remain reciting “explicit interface member mechanism”. Inherent in the term “explicit interface” as shown in Fig. 4, and see sec. 2.2, 2.3.

As per claim 9: Kramer further teaches *A method according to claim 1, wherein in addition to allowing the implementation of public interface members, said explicit interface member mechanism **enables the implementation** of private interface members.* The bold phase is relative

Art Unit: 2191

term that has the claim remain reciting “*explicit interface member mechanism*”. Inherent in the term “explicit interface” as shown in Fig. 4, and see sec. 2.2, 2.3.

As per claim 10: Kramer further teaches *A method according to claim 1, wherein said explicit interface member mechanism **enables** the implementation of a plurality of non-conflicting specific versions of a generic interface.* The bold phase is relative term that has the claim remain reciting “*explicit interface member mechanism*”. Inherent in the term “explicit interface” as shown in Fig. 4, and see sec. 2.2, 2.3.

As per claim 11: Kramer further teaches *A method according to claim 1, wherein the computer code is programmed according to an object-oriented programming language. See title “object-based”.*

As per claim 12: Kramer further teaches *A method according to claim 1, wherein an implementation of an explicit interface member is a method, property, event, or indexer declaration that references a fully qualified interface member name.* See sec. 2.2, 2.3.

As per claim 13 Kramer further teaches *A method according to claim 1, wherein the class names an interface in a base class list of the class that contains a member whose fully qualified name, type, and parameter types exactly match those of the implementation of the explicit interface member.* See sec. 2.2, 2.3, and refer to the used term “object-based” as seen in the title.

As per claim 14: Kramer further teaches *A method according to claim 1, wherein said explicit interface member mechanism includes an interface mapping mechanism that locates implementations of interface members in the class.* See Fig. 5, or refer to object binding.

Art Unit: 2191

As per claim 15: Kramer further teaches *A method according to claim 14, wherein said interface mapping mechanism locates an implementation for each member of each interface specified in a base class list of the class*. See Fig. 5, or refer to object binding.

As per claim 17: Kramer further teaches *A method according to claim 1, wherein it is not possible to override an explicit interface member implementation, but where an explicit interface member implementation calls another virtual method, derived classes are capable of overriding the implementation*. The limitation is indefinite. See sec. 2.2.

As per claim 18: Kramer further teaches *A method according to claim 1, wherein the class inherits an interface implementation **is permitted** to re-implement the interface by including the interface in the base class list of the software component*. The bold phase is a relative term, which has the claim remain reciting the inheritance of object-oriented. See sec. 2.4.

As per claim 19: Kramer further teaches *A method according to claim 1, wherein said explicit interface member mechanism **prevents conflict** among specific implementations of a generic interface*. The bold phase is a relative term, which has the claim remain reciting *explicit interface member mechanism*. See sec. 2.2.

As per claims 23-26, 28-37, 39-41: See related rationale addressed in claim 1-3, 6-15, 17-19.

As per claims 61-63, 66-75, 77-79: See related rationale addressed in claim 1-3, 6-15, 17-19.

Conclusion

5. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Ted T. Vo whose telephone number is (571) 272-3706. The examiner can normally be reached on 8:00AM to 4:30PM.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Wei Y. Zhen can be reached on (571) 272-3708.

The facsimile number for the organization where this application or proceeding is assigned is the Central Facsimile number 571-273-8300.

Any inquiry of a general nature or relating to the status of this application should be directed to the TC 2100 Group receptionist: 571-272-2100. Information regarding the status of

Art Unit: 2191

an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

TTV
May 23, 2010

/Ted T. Vo/
Primary Examiner, Art Unit 2191